

Binarizations in Random Number Generation

Sung-il Pae

Department of Computer Engineering, Hongik University
Seoul, Korea *

February 22, 2016

Abstract

Extracting procedures produce unbiased random bits from biased coin flips. *Binarizations* take inputs from an m -faced dice and produce bit sequences to be fed into a (binary) extracting procedure to obtain random bits, and this can be done in an entropy-preserving manner, without loss of information. Such a procedure has been proposed by Zhou and Bruck [1]. We discuss a family of such entropy-preserving processes that we call *complete binarizations*.

1 Introduction

An *m-extracting* procedure produces unbiased random bits using a sequence from an i.i.d. source over an alphabet $\{0, 1, \dots, m-1\}$, for example, a loaded dice with m faces, regardless of the probability distribution $\langle p_0, p_1, \dots, p_{m-1} \rangle$ of the source. When $m = 2$, the source is a biased coin with an unknown bias. The famous von Neumann trick takes input sequence of length 2 and returns random bits by the following rule [2]:

$$00 \mapsto \lambda, 01 \mapsto 0, 10 \mapsto 1, 11 \mapsto \lambda, \quad (1)$$

where λ indicates “no output.” Because $\Pr(01) = \Pr(10) = p_0p_1$, the resulting bit is unbiased, and the *rate*, the average number of output per input, is $p_0p_1 \leq 1/4$. Elias [3] and Peres [4] extend von Neumann’s trick by taking inputs of length $n \geq 2$ and returning more than one bit at a time; when $n = 2$, they coincide with the von Neumann’s method. Both methods are asymptotically optimal; as the input size n increases, the output rate approaches the information-theoretic upper bound $H(p_0)$, the Shannon entropy [5, 6].

*This work was supported in part by a Hongik University grant and the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. 2009-0077288).

Elias's method generalizes naturally to m -extracting procedures for each $m > 2$, and, in fact, the generalization is discussed briefly in Elias's original paper [3]. However, a similar generalization of Peres's method had been unknown for quite a while and was found only recently [7]. In the meanwhile, Zhou and Bruck proposed a very interesting method that transforms any binary extracting procedure into an m -extracting procedure [1]. For example, Peres method is turned into an m -extracting procedure via a simple process called "binarization." If the above-mentioned generalizations of Elias and Peres are to be called direct generalizations, their method is rather a meta-generalization. Moreover, the method is claimed to return an asymptotically optimal algorithm if the original binary algorithm is asymptotically optimal.

Such entropy-preserving processes are called *complete binarizations*. We discuss examples and a systematic way to construct a family of complete binarizations.

2 Extracting Procedures and Binarization

2.1 Extracting Procedures

Our dice X has m faces with values $0, 1, \dots, m-1$ with probability distribution $\langle p_0, \dots, p_{m-1} \rangle$. A sequence $x = x_1 \dots x_n \in \{0, 1, \dots, m-1\}^n$ is considered to be taken from n repeated throws of the dice. Summarized below are some necessary facts on extracting procedures. Refer to [8] and [7] for details.

Definition 1 ([4, 8]). A function $f: \{0, 1, \dots, m-1\}^n \rightarrow \{0, 1\}^*$ is m -extracting if for each pair z_1, z_2 in $\{0, 1\}^*$ such that $|z_1| = |z_2|$, we have $\Pr(f(x) = z_1) = \Pr(f(x) = z_2)$, regardless of the distribution $\langle p_0, \dots, p_{m-1} \rangle$.

Definition 2. A function $\Psi: \{0, 1, \dots, m-1\}^* \rightarrow \{0, 1\}^*$ is called an m -extracting procedure if its restriction on $\{0, 1, \dots, m-1\}^n$ is extracting, for every $n \geq 0$.

Sometimes, m is omitted, and a 2-extracting procedure is also called a binary extracting procedure.

Define Ψ_1 on $\{0, 1\}^2$ by the rule (1) and call it von Neumann function. Extend it by, for an empty string,

$$\Psi_1(\lambda) = \lambda,$$

for a nonempty even-length input,

$$\Psi_1(x_1 x_2 \dots x_{2n}) = \Psi_1(x_1 x_2) * \dots * \Psi_1(x_{2n-1} x_{2n}),$$

where $*$ is concatenation, and for an odd-length input, drop the last bit and take the remaining even-length bits. Then the resulting function Ψ_1 is a 2-extracting procedure. Of course, there are more interesting extracting procedures. Asymptotically optimal 2-extracting procedures like

Elias's [3, 9, 8] and Peres's [4, 10, 7] also extend von Neumann function but do not simply repeat it.

Denote by $S_{(n_0, n_1, \dots, n_{m-1})}$ the subset of $\{0, 1, \dots, m-1\}^n$ that consists of strings with n_i i 's. Then

$$\{0, 1, \dots, m-1\}^n = \bigcup_{n_0 + n_1 + \dots + n_{m-1} = n} S_{(n_0, n_1, \dots, n_{m-1})},$$

and each $S_{(n_0, n_1, \dots, n_{m-1})}$ is an *equiprobable* subset of elements whose probability of occurrence is $p_0^{n_0} p_1^{n_1} \dots p_{m-1}^{n_{m-1}}$. The size of an equiprobable set is given by a multinomial coefficient like

$$\binom{n}{n_0, n_1, \dots, n_{m-1}} = \frac{n!}{n_0! n_1! \dots n_{m-1}!}.$$

When $m = 2$, an equiprobable set $S_{(l, k)}$ is also written as $S_{n, k}$, where $n = l + k$, and its size can also be written as an equivalent binomial coefficient as well as the multinomial one:

$$\binom{n}{k} = \binom{n}{l, k}.$$

Extracting functions can be characterized using the concept of *multiset*. A multiset is a set with repeated elements; formally, a multiset M on a set S is a pair (S, ν) , where $\nu: S \rightarrow \mathbf{N}$ is a multiplicity function and $\nu(s)$ is called the *multiplicity*, or the number of *occurrences* of $s \in S$. The size $|M|$ of $M = (S, \nu)$ is $\sum_{s \in S} \nu(s)$. For multisets A and B , $A \uplus B$ is the multiset such that an element occurring a times in A and b times in B occurs $a + b$ times in $A \uplus B$. So $|A \uplus B| = |A| + |B|$, and the operation \uplus is associative.

When we write $x \in M = (S, \nu)$, it simply means that $x \in S$. However, when we use the expression " $x \in M$ " as an index, the multiplicity of the elements is taken into account. For example, for multisets A and B , the multiset $A \uplus B$ can be redefined as $\{x \mid x \in A \text{ or } x \in B\}$.

Definition 1 states that the image of an extracting function is multiple copies of $\{0, 1\}^N$, the exact full set of binary strings of various lengths N 's. For example, von Neumann procedure defined above sends $\{0, 1\}^6$ to 12 copies of $\{0, 1\}$, 6 copies $\{0, 1\}^2$, and one copy of $\{0, 1\}^3$.

Definition 3 ([7]). A multiset A of bit strings is extracting if, for each z that occurs in A , all the bit strings of length $|z|$ occur in A the same time as z occurs in A .

For multisets A and B of bit strings, define a new multiset $A * B = \{s * t \mid s \in A, t \in B\}$, and this operation is associative, too. If A and B are extracting, both $A * B$ and $A \uplus B$ are extracting. Denote by $f((C))$ the multiset $\{f(x) \mid x \in C\}$, or equivalently, $(f(C), \nu)$ with $\nu(z) = |f^{-1}(z) \cap C|$ for $z \in f(C)$. Note that $|f((C))| = |C|$. For a disjoint union $C \cup D$, we have $f((C \cup D)) = f((C)) \uplus f((D))$. With this notation, $\Psi_1((\{0, 1\}^6)) = 12 \cdot \{0, 1\} \uplus 6 \cdot \{0, 1\}^2 \uplus 1 \cdot \{0, 1\}^3$.

The following lemma reinterprets the definition of extracting function in terms of equiprobable sets.

Lemma 4 ([7]). A function $f: \{0, 1, \dots, m-1\}^n \rightarrow \{0, 1\}^*$ is extracting if and only if $f((S_{(n_0, n_1, \dots, n_{m-1})}))$ is extracting for each tuple $(n_0, n_1, \dots, n_{m-1})$ with $n_0 + n_1 + \dots + n_{m-1} = n$.

2.2 A Simple Example of Binarization

Let Ψ be a binary extracting procedure. Instead of a coin, suppose that we have a 4-face dice whose outcome X has the probability distribution $\langle p, q, r, s \rangle$. For a sample x of X , take the standard binary expansion x' , and let $\Phi_1(x)$ be its first bit and $\Phi_2(x)$ the second bit:

| x | $\text{Pr}(x)$ | x' | $\Phi_1(x)$ | $\Phi_2(x)$ |
|-----|----------------|------|-------------|-------------|
| 0 | p | 00 | 0 | 0 |
| 1 | q | 01 | 0 | 1 |
| 2 | r | 10 | 1 | 0 |
| 3 | s | 11 | 1 | 1 |

(2)

Then, $\Phi_1(X)$ and $\Phi_2(X)$ are Bernoulli random variables of distributions $\langle p+q, r+s \rangle$ and $\langle p+r, q+s \rangle$, respectively, so that we can feed to Ψ to obtain random bits. However, we lose information in the process; the bits from $\Psi(\Phi_1(X))$ and $\Psi(\Phi_2(X))$ are not independent to each other, and thus cannot be concatenated. Separately, neither $\Phi_1(X)$ nor $\Phi_2(X)$ cannot recover the information because their entropies are strictly smaller than $H(X)$.

2.3 An Entropy-Preserving Binarization

For a symbol $x \in \{0, 1, \dots, m-1\}$ and $1 \leq i \leq m-1$, consider

$$x^{(i)} = \begin{cases} 0, & x < i, \\ 1, & x = i, \\ \lambda, & x > i. \end{cases}$$

When $m = 6$, we have their values as follow:

| x | $\text{Pr}(x)$ | $x^{(1)}$ | $x^{(2)}$ | $x^{(3)}$ | $x^{(4)}$ | $x^{(5)}$ |
|-----|----------------|-----------|-----------|-----------|-----------|-----------|
| 0 | p_0 | 0 | 0 | 0 | 0 | 0 |
| 1 | p_1 | 1 | 0 | 0 | 0 | 0 |
| 2 | p_2 | λ | 1 | 0 | 0 | 0 |
| 3 | p_3 | λ | λ | 1 | 0 | 0 |
| 4 | p_4 | λ | λ | λ | 1 | 0 |
| 5 | p_5 | λ | λ | λ | λ | 1 |

(3)

For $x = x_1 \dots x_n \in \{0, 1, \dots, m-1\}^n$, define $x^{(i)} = x_1^{(i)} * \dots * x_n^{(i)}$. So for a sequence x of length n , $x^{(i)}$ is a binary sequence of length at most n . We will show that, for a binary extracting procedure Ψ , a function $\Psi' : \{0, 1, \dots, m-1\}^n \rightarrow \{0, 1\}^*$, defined by

$$\Psi'(x) = \Psi(x^{(1)}) * \dots * \Psi(x^{(m-1)}),$$

is m -extracting.

Now, for a subset $S \subset \{0, 1, \dots, m-1\}^n$, let $S^{(i)} = \{x^{(i)} \mid x \in S\}$. Then, for an equiprobable subset $S \in \{0, 1, \dots, m-1\}^n$, we can see that $S^{(i)}$ is another equiprobable set in $\{0, 1\}^{n'}$. For example, for $S = S_{(1,2,1)}$, observe that

| x | $x^{(2)}$ | $x^{(1)}$ |
|------|-----------|-----------|
| 0112 | 0001 | 011 |
| 0121 | 0010 | 011 |
| 0211 | 0100 | 011 |
| 1012 | 0001 | 101 |
| 1021 | 0010 | 101 |
| 1102 | 0001 | 110 |
| 1120 | 0010 | 110 |
| 1201 | 0100 | 101 |
| 1210 | 0100 | 110 |
| 2011 | 1000 | 011 |
| 2101 | 1000 | 101 |
| 2110 | 1000 | 110 |

and we can conclude that

$$\begin{aligned} S^{(1)} &= 4 \cdot S_{(1,2)} \\ S^{(2)} &= 3 \cdot S_{(3,1)}. \end{aligned}$$

Note that

$$|S_{(1,2,1)}| = \frac{4!}{1!2!1!} = \frac{3!}{1!2!} \times \frac{4!}{3!1!} = |S_{(1,2)}| \times |S_{(3,1)}|.$$

In fact, we have a stronger claim:

Lemma 5 (Structure Lemma). *The mapping $\Phi: x \mapsto (x^{(1)}, \dots, x^{(m-1)})$ gives a one-to-one correspondence between equiprobable subset $S = S_{(n_0, n_1, \dots, n_{m-1})}$ and $S^{(1)} \times \dots \times S^{(m-1)}$, where*

$$S^{(i)} = S_{(n_0 + \dots + n_{i-1}, n_i)}.$$

Theorem 6. *For a binary extracting procedure Ψ , a function $\Psi' : \{0, 1, \dots, m-1\}^n \rightarrow \{0, 1\}^*$ defined by*

$$\Psi'(x) = \Psi(x^{(1)}) * \dots * \Psi(x^{(m-1)}),$$

is m -extracting for each n .

Proof. By Lemma 5, for an equiprobable set S , each $S^{(i)}$ is equiprobable and thus $\Psi(S^{(i)})$ is extracting. Moreover, by the same lemma, $\Psi'(S) = \Psi(S^{(1)}) * \dots * \Psi(S^{(m-1)})$. Since each $\Psi(S^{(i)})$ is extracting, $\Psi'(S)$ is extracting. Theorem follows by Lemma 4. \square

The distribution of $X^{(i)}$ is $\langle p^{(i)}, q^{(i)} \rangle$, where

$$p^{(i)} = \frac{p_0 + \dots + p_{i-1}}{p_0 + \dots + p_i}, \quad q^{(i)} = \frac{p_i}{p_0 + \dots + p_i},$$

and the $X^{(i)}$ has an output with probability $p_0 + \dots + p_i$. Therefore, if Ψ is asymptotically optimal, then the rate of $\Psi(x^{(i)})$ converges to $(p_0 + \dots + p_i)H(p^{(i)})$ as the input size increases.

Lemma 7. *The weighted sum of the entropies of $X^{(i)}$ equals the entropy of X . That is,*

$$(p_0 + p_1)H(p^{(1)}) + (p_0 + p_1 + p_2)H(p^{(2)}) + \dots + H(p^{(m-1)}) = \sum_{i=1}^{m-1} \lg p_i.$$

Therefore, the rate of Ψ' approaches to the entropy of X as the input size tends to infinity. So we have:

Theorem 8. *If a binary extracting procedure Ψ is asymptotically optimal, then the m -extracting procedure $x \mapsto \Psi'(x) = \Psi(x^{(1)}) * \dots * \Psi(x^{(m-1)})$ is asymptotically optimal.*

Proofs of Lemmas 5 and 7 are given in a more general setting in Section 3.

2.4 Zhou-Bruck Binarization

The following method was proposed by Zhou and Bruck [1]. For $x \in \{0, 1, \dots, m-1\}$, let x' be the $\lceil \lg m \rceil$ -bit binary expansion of x , and also for $\alpha \in \{0, 1\}^*$, let

$$x^\alpha = \begin{cases} a, & \text{if } \alpha a \text{ is a prefix of } x', \\ \lambda, & \text{otherwise.} \end{cases}$$

That is, x^α is the bit that immediately follows α in x' . For example, when $m = 6$, we have

| x | x' | x^λ | x^0 | x^1 | x^{00} | x^{01} | x^{10} |
|-----|------|-------------|-----------|-----------|-----------|-----------|-----------|
| 0 | 000 | 0 | 0 | λ | 0 | λ | λ |
| 1 | 001 | 0 | 0 | λ | 1 | λ | λ |
| 2 | 010 | 0 | 1 | λ | λ | 0 | λ |
| 3 | 011 | 0 | 1 | λ | λ | 1 | λ |
| 4 | 100 | 1 | λ | 0 | λ | λ | 0 |
| 5 | 101 | 1 | λ | 0 | λ | λ | 1 |

(4)

As with the example (3), analogous lemmas and theorem hold, and $x \mapsto \Psi'(x) = \Psi(x^\lambda) * \dots * \Psi(x^{1\dots 1})$ is an asymptotically optimal m -extracting procedure if Ψ is asymptotically optimal.

3 Complete Binarizations

Given a function $\phi: \{0, 1, \dots, m-1\} \rightarrow \{0, 1, \lambda\}$, $\phi(X)$ is a Bernoulli random variable with distribution $\langle p, q \rangle$, where

$$p = \sum_{\phi(i)=0} p_i/s, \quad q = \sum_{\phi(i)=1} p_i/s, \quad \text{and } s = \sum_{\phi(i) \neq \lambda} p_i.$$

Extend ϕ to $\{0, 1, \dots, m-1\}^n$, by letting, for $x = x_1 \dots x_n$, $\phi(x) = \phi(x_1) * \dots * \phi(x_n)$. Then, for an equiprobable set $S = S_{(n_0, \dots, n_{m-1})}$,

$$\phi(S) = S_{(l, k)},$$

where

$$l = \sum_{\phi(i)=0} n_i, \quad k = \sum_{\phi(i)=1} n_i.$$

A *binarization* takes a sequence over $\{0, 1, \dots, m-1\}$ and outputs several binary sequences that are to be separately fed into a binary extracting procedure and then concatenated together to obtain random bits.

Definition 9. A collection of mappings $\Phi = \{\Phi_i : \{0, 1, \dots, m-1\} \rightarrow \{0, 1, \lambda\} \mid i = 1, \dots, M\}$ is called a *binarization* if, when extended to $\{0, 1, \dots, m-1\}^n$, given a 2-extracting procedure Ψ , the mapping $x \mapsto \Psi'(x) = \Psi(\Phi_1(x)) * \dots * \Psi(\Phi_M(x))$ is an m -extracting function. Here, each Φ_i is called a *component* of Φ , and we often write $\Phi(x) = (\Phi_1(x), \dots, \Phi_M(x))$. For an asymptotically optimal 2-extracting procedure Ψ , if the resulting Ψ' is asymptotically optimal, then Φ is called a *complete binarization*.

Now, for a function $\phi: \{0, 1, \dots, m-1\} \rightarrow \{0, 1, \lambda\}$, let

$$\begin{aligned} \text{supp}_0(\phi) &= \{x \mid \phi(x) = 0\} \\ \text{supp}_1(\phi) &= \{x \mid \phi(x) = 1\} \\ \text{supp}(\phi) &= \{x \mid \phi(x) \neq \lambda\} = \text{supp}_0(\phi) \cup \text{supp}_1(\phi), \end{aligned}$$

and call them 0-support, 1-support, and support of ϕ , respectively. Call ϕ *degenerate* if its 0-support or 1-support is empty so that $\phi(X)$ is a degenerate Bernoulli process.

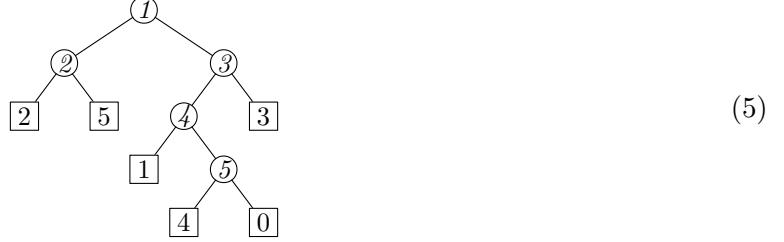
Consider a binary tree with m external nodes labeled uniquely with $0, 1, \dots, m-1$. For an internal node v define a function $\phi_v: \{0, 1, \dots, m-1\} \rightarrow \{0, 1, \lambda\}$ as follows:

$$\phi_v(x) = \begin{cases} 0, & \text{if } x \in \text{leaf}_0(v), \\ 1, & \text{if } x \in \text{leaf}_1(v), \\ \lambda, & \text{otherwise.} \end{cases}$$

where $\text{leaf}_0(v)$ ($\text{leaf}_1(v)$, respectively) is the set of external nodes on the left (right, respectively) subtree of v . Since there are exactly $m-1$ internal nodes, we uniquely name them with $1, \dots, m-1$,

with 1 the root node, and the corresponding functions $\Phi_1, \dots, \Phi_{m-1}$. Also define $a(i)$ ($b(i)$, respectively) to be i 's left (right, respectively) child if exists, 0 otherwise. Call such trees m -binarization trees.

For example,

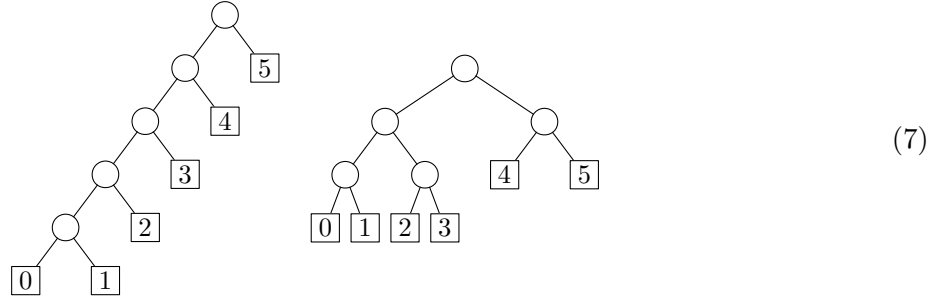


defines the following functions:

| x | $\Phi_1(x)$ | $\Phi_2(x)$ | $\Phi_3(x)$ | $\Phi_4(x)$ | $\Phi_5(x)$ |
|-----|-------------|-------------|-------------|-------------|-------------|
| 0 | 1 | λ | 0 | 1 | 1 |
| 1 | 1 | λ | 0 | 0 | λ |
| 2 | 0 | 0 | λ | λ | λ |
| 3 | 1 | λ | 1 | λ | λ |
| 4 | 1 | λ | 0 | 1 | 0 |
| 5 | 0 | 1 | λ | λ | λ |

and $a(1) = 2$, $a(2) = 0$, $a(3) = 4$, etc., and $b(1) = 3$, $b(2) = 0$, $b(3) = 0$, etc.

The binarizations (3) and (4) corresponds to the following trees with appropriate associations to the nodes:



Note that in (4) we have $5 = m - 1$ components left after the degenerate x^1 is removed.

Theorem 10. *Functions defined by an m -binarization tree make a complete binarization.*


As with Theorem 6, this theorem follows from Lemma 4 and the two lemmas, Structure Lemma and Entropy Lemma, given below.

Lemma 11 (Structure Lemma). *Let $\Phi = \{\Phi_1, \dots, \Phi_{m-1}\}$ be the set of functions defined by an m -binarization tree. Then the mapping $\Phi: x \mapsto \Phi(x) = (\Phi_1(x), \dots, \Phi_{m-1}(x))$ gives a one-to-one correspondence between an equiprobable subset $S = S_{(n_0, n_1, \dots, n_{m-1})}$ and $\Phi_1(S) \times \dots \times \Phi_{m-1}(S)$.*



Proof. Let $S_i = \Phi_i(S)$, for $i = 1, \dots, m-1$. First, observe that the sizes of two sets match, that is:

$$|S_1 \times \dots \times S_{m-1}| = \binom{n}{n_0, \dots, n_{m-1}} = |S|, \quad (8)$$


using induction and the following cases as bases:

In case , we have $\phi(S) = S_{(n_i, n_j)}$, and thus

$$|\phi(S)| = \binom{n_i + n_j}{n_i, n_j}.$$

In case , (and similarly with , where $k = |\text{supp}_1(\phi)| = |\text{supp}(\varphi)|$, $l' = |\text{supp}_0(\varphi)|$ and $k' = |\text{supp}_1(\varphi)|$, we have $\phi(S) = S_{(n_i, k)}$, $\varphi(S) = S_{(l', k')}$, and $k = l' + k'$. So

$$|\phi(S) \times \varphi(S)| = \binom{n_i + k}{n_i, k} \binom{k}{l', k'} = \binom{n_i + k}{n_i, l', k'}.$$

In case , where $\phi(S) = S_{(l, k)}$, $\phi_a(S) = S_{(l_a, k_a)}$, and $\phi_b(S) = S_{(l_b, k_b)}$, we have $l = l_a + k_a$, $k = l_b + k_b$, and

$$\begin{aligned} |\phi(S) \times \phi_a(S) \times \phi_b(S)| &= \binom{l + k}{l, k} \binom{l}{l_a, k_a} \binom{k}{l_b, k_b} \\ &= \binom{l + k}{l_a, k_a, l_b, k_b}. \end{aligned}$$

All the l 's with $|\text{supp}_0(\Phi_i)| \geq 2$ and k 's with $|\text{supp}_1(\Phi_i)| \geq 2$ cancel out so that we have (8).

Since $\Phi(S) \subset S_1 \times \dots \times S_{m-1}$ and $|S| = |S_1 \times \dots \times S_{m-1}|$, now we only need to see that Φ is injective on S (in fact on $\{0, 1, \dots, m-1\}^n$).

An m -binarization tree is also a code tree for a prefix code over the alphabet $\{0, 1, \dots, m-1\}$. For example, a symbol '4' is encoded by codeword '1010' by the tree (5). Let $\text{code}(x)$ be the code for $x = x_1 \dots x_n \in \{0, 1, \dots, m-1\}^n$ determined by the given m -binarization tree. The following procedure constructs binary sequence code $(x) = c_1 \dots c_K$ for $y = (y_1, \dots, y_{m-1}) = \Phi(x) = (\Phi_1(x), \dots, \Phi_{m-1}(x))$:

```

 $k \leftarrow 1$ 
for  $i \leftarrow 1$  to  $n$       (for each  $x_i$ ,  $i = 1, \dots, n$ )
   $j \leftarrow 1$ 
  while  $j \neq 0$ 
     $c_k \leftarrow$  first bit of  $y_j$ , and remove it from  $y_j$ 
    if  $c_k = 0$ ,  $j \leftarrow a(j)$ ; else  $j \leftarrow b(j)$ 
     $k \leftarrow k + 1$ 

```

Conversely, given $\text{code}(x)$, the following procedure finds $\Phi(x) = (\Phi_1(x), \dots, \Phi_{m-1}(x))$:

```

 $k \leftarrow 1$ 
while  $k \leq K$ 
   $j \leftarrow 1$ 
  while  $j \neq 0$ 
     $j_k \leftarrow y_j * c_k$ 
    if  $c_k = 0$ ,  $j \leftarrow a(j)$ ; else  $j \leftarrow b(j)$ 
   $k \leftarrow k + 1$ 

```

This establishes a one-to-one correspondence $\text{code}(x) \mapsto \Phi(x)$. Since $\text{code}(x)$ is unique, Φ is injective. \square

The process $\Phi_i(X)$ has an output with probability $\pi_i = \sum_{j \in \text{supp}(\Phi_i)} p_j$, and its distribution is $\langle P_i, Q_i \rangle$, where

$$P_i = \sum_{j \in \text{supp}_0(\Phi_i)} p_j / \pi_i, \quad Q_i = \sum_{j \in \text{supp}_1(\Phi_i)} p_j / \pi_i.$$

Note that P_i and Q_i are recursively given by π_i 's: $\pi_1 = 1$ and

$$P_i = \pi_{a(i)} / \pi_i, \quad Q_i = \pi_{b(i)} / \pi_i.$$

Lemma 12 (Entropy Lemma). *The entropies of $\Phi_i(X)$ weighted by the probability π_i sum up to the entropy of X :*

$$\sum_{i=1}^{m-1} \pi_i H(P_i) = H(X).$$

Proof. Similar cancellations occur as in the proof of the Structure Lemma. See Lemma E in Section 6.2.2 of [11] for a more general statement and proof. \square

4 Remarks

1. If we omit non-degenerate components, say $\Phi_{i_1}, \dots, \Phi_{i_k}$, from a complete binarization, then the corresponding structure lemma gives, for each equiprobable set S , instead of one-to-one, a one-to- $|\Phi_{i_1}(S)| \times \dots \times |\Phi_{i_k}(S)|$ map, which still gives a binarization but loses entropy. Hence the name “complete.”
2. The author believes that essentially all possible complete binarizations are given by a binarization tree.

References

- [1] H. Zhou and J. Bruck, “A universal scheme for transforming binary algorithms to generate random bits from loaded dice,” *CoRR*, vol. abs/1209.0726, 2012. [Online]. Available: <http://arxiv.org/abs/1209.0726>
- [2] J. von Neumann, “Various techniques for use in connection with random digits. Notes by G. E. Forsythe,” in *Monte Carlo Method, Applied Mathematics Series*. U.S. National Bureau of Standards, Washington D.C., 1951, vol. 12, pp. 36–38, reprinted in von Neumann’s *Collected Works* 5 (Pergammon Press, 1963), 768–770.
- [3] P. Elias, “The efficient construction of an unbiased random sequence,” *The Annals of Mathematical Statistics*, vol. 43, no. 3, pp. 865–870, 1972.
- [4] Y. Peres, “Iterating von Neumann’s procedure for extracting random bits,” *Annals of Statistics*, vol. 20, no. 1, pp. 590–597, 1992.
- [5] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana: The University of Illinois Press, 1964.
- [6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunications. New York, NY, USA: John Wiley & Sons, 1991.
- [7] S. Pae, “A generalization of Peres’s algorithm for generating random bits from loaded dice,” *IEEE Transactions on Information Theory*, vol. 61, no. 2, 2015.
- [8] S. Pae and M. C. Loui, “Randomizing functions: Simulation of discrete probability distribution using a source of unknown distribution,” *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 4965–4976, November 2006.
- [9] —, “Optimal random number generation from a biased coin,” in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2005, pp. 1079–1088.
- [10] S. Pae, “Exact output rate of Peres’s algorithm for random number generation,” *Inf. Process. Lett.*, vol. 113, no. 5-6, pp. 160–164, 2013.
- [11] D. E. Knuth, *The Art of Computer Programming, Sorting and Searching*, 2nd ed. Addison-Wesley, 1998, vol. 3.